2009

May the Source be with You: Exploring the Efficiency of Open Source Techniques

Will Dickey
Trinity College

May the Source be with You: Exploring the Efficiency of Open Source Techniques

Will Dickey

In many ways software is like your car. Under the hood, your car is made up of hundreds of different parts and mechanisms that the manufacturer decided to install. Most people do not care to know the complex structure of the car, but just care if it runs properly or not. In computers, the parts under the hood are called the source code: commands that decide the functions of a computer. Like with a car engine, most people don't care to change the properties of their source code, simply because they don't need to understand how it works. They may pay a mechanic to change their car's oil, so only the mechanic needs to open the hood. Proprietary software is like a car with the hood welded shut. Only the software company can see what's really happening and make changes they think the consumer will benefit from.[1]

If anyone can open the hood, anyone can compete in the business of changing oil. In computers this open-hood approach is called the open-source model. Perhaps someone's neighbor is also a mechanic and offers to change the oil for less money. Now it matters if who can open the hood. That neighbor may even be a very talented mechanic and will upgrade your car for the cost of parts. With proprietary software, this kind of exchange isn't possible. The open-source model allows anyone to open the hood and modify the structure, so to speak. People are free to share changes and upgrades with their neighbors. Eventually, they might collaborate with whole groups of people to develop a better engine. Of course, software isn't actually like a car, but the principles are the same. Open-source production allows large groups of people to communicate and share information in order to develop more efficient results than the proprietary model. With new technology, the model of production has shifted its focus from the commercial exchange of property to one centered on the sharing of information, such as that in financial services, software, and science. The Internet has brought about more cultural creation and more integration of different economic sectors. Attaching this new model to a communications environment built on cheap processors, connected to billions of people, begins a new phenomenon of social networking.

We are in the middle of a drastic sorting of property and information. With the Internet and open-source software, the exchange of intellectual property is creating a maelstrom of legal confusion and business opportunities. Commons-based peer production has harnessed the new ability to network with multitudes of people in order to create newer and better products. In this paper, I will look back on the history of open source production and explore how it has provided a new basis for today's industries.[2]

Over the past two decades, there has been a surge in interest of open-source development especially in the software market. The idea for transparency in computing can be traced back to Richard Stallman. During the 1980s Stallman, who was working at MIT, was troubled by the industry standard of proprietary software.[3] Programmers were

---

[1] Stallman, "Free Software, Free Society: Selected Essays of Richard M. Stallman."
[2] Benkler, "Commons-based peer production and virtue.."
[3] Moore, *Revolution OS*.

paid for their labor but could not control their code. Users could buy a copy of a program but couldn't change or distribute it. Without the source code, a program was inflexible because complicated user licenses prohibited users from seeing the source code.[4] Stallman wanted all users to have true ownership of their computers through "free" software-- software which was free in the sense of free speech, not free beer. He developed a project to create a completely free operating system called GNU. In order to make sure that companies didn't steal his software and license the code, he created the General Public License (GPL), which prevented anyone from restricting the use of the GNU software. [5] Since Stallman's ideology was so radical, it scared away many companies from participating in his project because they saw no real mainstream or commercial value. Programmer Eric Raymond states in his important essay, "Bazaar," that "[if] you want to change the world, you have to co-opt the people who write the big checks."[6]

Another problem with the GNU project was that it had no kernel; a key component to all operating system that connects the core processes of a computer together. In the early 1990s, Linus Torvalds was impatient with the development of GNU and wanted to make a new operating system for his own personal computer.[7] Using the software provided under the GPL, he created the Linux kernel and asked other developers to join him in expanding the software. With the collaboration of thousands of developers connected through the Internet, the project exploded and Linux quickly became the most popular free software on the market. As firms began to see its functionality, a new open-source model of production emerged in the software world. During the 1990s, many startup companies adopted the Linux model of open-source production. As the Internet burst into the mainstream, open source projects like Sendmail and Apache grew in popularity.[8] Today, Apache is the dominant Web-server solution, and it runs under Linux programming. For this and other reasons, the rapid growth of the Internet and the rising popularity of open source software are interdependent. Linux, Apache, and other open-source software could outperform costly proprietary UNIX or Microsoft solutions, giving them a competitive advantage for Web-based companies.[9]

The difference in production structures helps to explain this phenomenon of openness in both technology and co-operation. In 1997, Raymond wrote his essay," The In Raymond's so-called cathedral, a leader sets the goals, encourages programmers to participate, uses monetary rewards, and controls the product and its secrets. In the bazaar, a leader shares a vision, invites programmers, rewards their contributions with fame and gratitude, and shares the product as open source.[10] Raymond emphasizes the fame reward, using the theory of "reputation culture" to explain programmers' eagerness to work hard on code and then give it away.[11]

---

[4] "Arguments about open source."
[5] Moore, *Revolution OS*.
[6] Ibid.
[7] "Arguments about open source."
[8] Scacchi, "Free/open source software development."
[9] Gupta, "Open Source Economics Driving Web 2.0 Innovation."
[10] Raymond, "The Cathedral and the Bazaar."
[11] Lerner, "The Simple Economics of Open Source."

The success of the open-source model led some proprietary companies to reconsider their own structures of business. One of the most famous of these examples is Web browser Netscape's shift to Mozilla Firefox. In the late 1990s Netscape's Communicator and Microsoft's Internet Explorer were locked in a fierce battle for market share. To try and gain a competitive advantage, Netscape announced that it would switch to an open-source model, meaning that it would develop its code through tools that could be shared with multiple development teams and a volunteer community.[12] This was a drastic change for Netscape's employees, who would be put in a fish-bowl-like environment where their actions would become transparent to all other members of the project. But with this new open-source technique, more code could be written to support more frequent releases of new Mozilla platforms. Since 2004, Mozilla has averaged over 250,000 downloads per day and has reached over 25% of the market share, a testament to the efficiency and work ethic of the new product.[13]

Some will argue that the Mozilla project was aided by other proprietary companies like AOL and thus wasn't an official open-source venture. But the fundamentals of open source stayed genuine. The "fish-bowl" analogy is important in all open-source models, because it demonstrates the real "openness" of the operation.[14] Mozilla was special because it was the result of a change from proprietary software to open source. Along with this change in both philosophy and production, the converted Netscape employees demonstrated that being transparent with each other allowed them to create a better end product for the consumer.[15] Not only was the successful software open source, but the population of employees also demonstrated that they could cooperate effectively through a Creative Commons methodology, a methodology that emphasizes peer cooperation and transparency to create better products.

The factor that separates all open source projects from most propriety enterprises is this sharing of information and collaboration among large groups. According to Yochai Benkler, FOSS projects use a production tactic that he called commons-based peer production, "a socio-economic system of production that is currently emerging in the digitally networked environment."[16] The collaboration among large groups of individuals sometimes in the order of tens or even hundreds of thousands is assisted by the technical infrastructure of the Internet. Unlike the proprietary model, "groups work to provide information, knowledge or cultural goods without relying on either market pricing or managerial hierarchies in order to coordinate their common enterprise."[17]

In recent years, there have been many successful community-based open-source productions. This includes successful software companies like Linux, Apache, and Firefox, but also projects that are not directly related to the fabrication of source code. Take SETI@home for example. The project uses millions of Internet-connected computers to sort through telescope imagery in the search for extra-terrestrial life. Each

---

[12] Baker, "The Mozilla Project: Past and Future."
[13] Faldetta, "The Content of Freedom in Resources: The Open Source Model."
[14] Baker, "The Mozilla Project: Past and Future."
[15] Ibid.
[16] Benkler, "Commons-based peer production and virtue.."
[17] Boyle, "The Second Enclosure Movement and the Construction of the Public Domain."

voluntary participant downloads a small program that runs during their computer's sleep cycles to break down collected data. According to statistics on the site, at the end of 2003, the project had accumulated over 4.5 million users from 225 countries.[18] With all this networking power, the "commons based computer" is twice as fast as the fastest supercomputer in the world.

The SETI@home project is a good example of multiple units combining to reach a greater goal. But open-source and Creative -Commons productions are inherently based on individual intellectual donation.[19] The world's most popular encyclopedia, Wikipedia, is a perfect example of people combining their knowledge in an open-source environment. The site, which is the sixth most visited in the world, contains millions of articles written and edited by a large community of contributors. Not surprisingly, in a battle of the encyclopedias, Wikipedia beat out its proprietary competitor Britannica in terms of errors per page.[20] The most important aspect of Wikipedia is that it encourages online user contribution and interaction through its open-source infrastructure. The goal of Wikipedia is not to create a set of opinion pieces, rather to create an "encyclopedia" based on factual knowledge acquired by human effort. [21]

So what are the motives for open-source collaboration? Think of the proprietary software system as a feudal market. In the feudal system of the middle Ages, farmers did not truly own their land; the lord of the region owned all the land, and farmers harvested crops in return for payment by the lord.[22] Technology companies like Microsoft employ a similar model, in which workers produce code that is owned by the company, and are not free to change what part of the code they are working on or use the code for other purposes without permission from the company.[23] If software is transparent, anyone in the company can see what changes are made and what software is added or subtracted. Like with Wikipedia, when a large community is added to a project, the number of eyes on the software multiplies, so that, for example, bugs and errors are fixed faster. Linus' Law states: "Given enough eyeballs, all bugs are shallow".[24] The larger the number of people who can potentially work on open- source projects, the larger the resource reservoir becomes. The internal structure of open-source development is shaped in a hierarchy similar to the proprietary model, but money, power, or authority does not shape each terrace, rather it is based on the magnitude of individual achievement and reputation. [25]

Participation in open source from the individual's perspective is determined by the future benefits they expect to receive. Of course monetary payment is a strong initiative, but a programmer working to fix a bug or a Wikipedia contributor also receives a payment of self-satisfaction and reputation through their efforts within the

---

[18] Benkler, "Commons-based peer production and virtue.."
[19] Benkler, *The Wealth of Networks*.
[20] Halavais and Lackaff, "An Analysis of Topical Coverage of Wikipedia."
[21] Scacchi, "Free/open source software development."
[22] Gupta, "Open Source Economics Driving Web 2.0 Innovation."
[23] "Arguments about open source."
[24] Raymond, "The Cathedral and the Bazaar."
[25] Boyle, "The Second Enclosure Movement and the Construction of the Public Domain."

community.[26] Benkler proposes the philosophical theory that programmers participate in open-source projects because they are acting out virtues such as benevolence and altruism. He states that participants in a commons-based peer effort cooperate, build upon the work of others, contribute time, effort and expertise to create and enhance a public good. The same can be said for participants in commercial-open source ventures who are also working transparently with others to create a better product.[27] The weakness point of open-source development until recently was the speed of the exchange of knowledge among peers. But now, with technology and servers as cheap as ever, the spread of information is extremely efficient, so the cost to the developer is minimal.

In open source, a programmer is his or her own boss and can take full responsibility for the success or failure of a task. Programmers in typical commercial projects, by contrast, need to work with, or around, their supervisors. Therefore, the individual contribution is harder to measure.

The real advantage of open source is in the delayed incentives, where the visibility of the programmer's contribution counts most. Open-source projects measure individual performance better. In a commercially created program, outsiders can't really tell who did what. Open source is different. As Lerner writes, "[o]utsiders are able to see not only what the contribution of each individual was and whether that component 'worked,' but also whether the task was hard, if the problem was addressed in a clever way, whether the code can be useful for other programming tasks in the future." Proprietary software companies such as Microsoft keep different programming sections separated so that employees are isolated from knowing the code that is produced even *within* their own company.[28] Such restrictions inhibit the company's dynamic and ability to transfer developer talent.

Finally, in open source, people have greater flexibility when moving from project to project, building up knowledge and tools as they go. By contrast, in commercial firms people are restricted by proprietary code specific to that firm. In many ways, they have to start all over again when they switch jobs. In their working paper, activists of open business point out that people in open source can use their projects as a "port of entry."[29] For example, a systems administrator at a small open-source project can signal her talent to many people in a position to influence her future career: colleagues, prospective employers and, especially, venture capitalists.[30]

Commercial firms have not failed to notice the success of open-source projects. Their strategies for capturing some of this energy usually fall into one of two strategies. One of these is the reactive strategy, where commercial firms try to package paid services and products onto open source programs, to fill a niche.[31] These services and products are either not provided at all by open source or are not handled very efficiently. "The

---

[26] Riehle, "The Economic Motivation of Open Source Software: Stakeholder Perspectives."
[27] "Arguments about open source."
[28] Riehle, "The Economic Motivation of Open Source Software: Stakeholder Perspectives."
[29] Ibid.
[30] Ibid.
[31] Faldetta, "The Content of Freedom in Resources: The Open Source Model."

company expects to … boost its profit on a complementary segment," Lerner writes.[32] Realizing the potential of open-source, in 2004, IBM released the source code to various applications with the intent of improving it and gaining recognition through the open-source community. This exchange benefits both parties because Linux will be able to adopt more Microsoft compatible technology and Microsoft will be able to improve its product through open-source collaboration.

In the second strategy, companies embrace the open-source movement by releasing some of their own proprietary code, in the hopes that this will lead to greater value down the road thanks to new kinds of cooperation. This type of release is similar to giving away the razor to sell more razor blades.[33] Just recently Microsoft released 20,000 lines of code to the Linux community for inclusion in the Linux program tree.[34]

The open-source movement leaves several questions for economists to contemplate in the future. How will the divisions of projects into components help or hurt open source? The success of an open-source project seems dependent on the ability to break down the project into distinct components. Speculators also wonder whether open-source projects can handle so many contributors jumping on the bandwagon. And finally, can open-source projects expect to live longer than commercial ones? Since open-source code is freely available, programs can survive as long as public interest is sufficient. But fads erupt in open source as in any other field, and developers who are attracted to high-profile projects, could be paying a higher opportunity cost if they leave a proprietary job.[35]

On a more ethical side, open-source supporters warn that proprietary software leads to monopolies through “lock-in” and other unfair business practices. They argue that proprietary companies can exclude competitors by enticing clients to purchase and upgrade their software with little alternative. This practice suppresses innovation and prevents open-source companies from reaching new customers and improving their products. Virtual monopolies have less pressure to innovate and little pressure to lower prices. [36]

In an example regarding the production of drugs, the patents on pharmaceuticals have limited durations. Therefore, new proprietary drugs can eventually be replicated as generics. Consumers may depend on the drugs, but when they become generic commodities, prices fall. Consumers have more choice and may pay less. Some open-source proponents want software to be like generic drugs. Proprietary software may have features or other value added, but the generic, open-source software would be a reasonable, economical choice. [37]

In modern times we are seeing fewer start-up open-source efforts while more proprietary companies are adopting already established open-source projects. Much of

---

[32] Gupta, “Open Source Economics Driving Web 2.0 Innovation.”
[33] Lerner, “The Simple Economics of Open Source.”
[34] McGuire, “Eyes Wide Open: Software Vendors Embrace Open Source.”
[35] Benkler, *The Wealth of Networks*.
[36] “Arguments about open source.”
[37] Faldetta, “The Content of Freedom in Resources: The Open Source Model.”

this is due to the increased accessibility and efficiency of the Internet and the popularity of Linux and Apache as open-source platforms. In 2003, Verizon saved $6 million in equipment costs by switching its programmers to Linux computers. Also from 2002-2003, Hewlett Packard's revenues increased 40% with the use of the Linux servers. Online vendors are also benefitting. Amazon.com allegedly saved $17 million in technology and telecommunications costs in one quarter, due in part to their migration to Red Hat Linux. [38]

The old business model of specialization and distributed labor has met its 2.0 version. Now, with open-source, ideas are connected for the purpose of developing new products and software. The change of the White House's Web site from proprietary software to open source is a microcosm of the global ideological change in technology and business and a boost to the whole movement. Its move to Drupal shows that the White House believes that open source is safe, cheap, and encourages the users to develop software as they see fit, [39] and that the collaborative development model allows for faster review and testing, making more efficient and innovative products.

There are many answers still to be discovered about open-source software and the open-source production model. But whether we like it or not, the way we live and work is becoming increasingly transparent. Facebook has now attracted over 350 million users, and although it is not technically an open-source website, it is a microcosm of the way the Internet and social networking have transformed our lives. Businesses like Google, IBM, and HP have realized the benefits of tapping into this new forum of connectivity where employees and users can share ideas and information freely. Agricultural and biotechnology companies that have no market share in technology have harnessed the new efficiency of communication to harvest and create better and more reliable products. A new model, better yet, a new lifestyle of open source has arrived.

BIBLIOGRAPHY

"Arguments about open source." *Open Options* (September 30, 2005). http://www.netc.org/openoptions/pros_cons/comparing.html.

Baker, Mitchell. "The Mozilla Project: Past and Future." *Open Sources 2.0: the continuing evolution*. http://www.archive.org/stream/opensources2.000diborich#page/n43/mode/2up.

Benkler, Yochai. "Commons-based peer production and virtue.." *Journal of Political Philosophy* 14, no. 4 (2006): 394-419.

———. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006. http://portal.acm.org/citation.cfm?id=1214507.

[38] Walker, "The Future of Open Source in Government."
[39] Ibid.

Boyle, James. "The Second Enclosure Movement and the Construction of the Public Domain." *Duke University School of Law* 66, no. 1 (Winter 2003): 33-74.

Faldetta, Guglielmo. "The Content of Freedom in Resources: The Open Source Model." *Journal of Business Ethics* 39, no. 1 (August 2002): 179-188.

Gupta, Jitendra. "Open Source Economics Driving Web 2.0 Innovation." *Read Write Web*. http://www.readwriteweb.com/archives/open_source_economics.php.

Halavais, Alexander, and Derek Lackaff. "An Analysis of Topical Coverage of Wikipedia." *Journal of Computer-Mediated Communication* 13, no. 2 (1, 2008): 429-440.

Lerner, Josh. "The Simple Economics of Open Source." *Harvard Business School* (December 29, 2000). www.nber.org/papers/w7600.

McGuire, Craig. "Eyes Wide Open: Software Vendors Embrace Open Source." *Digital Media Buzz*, September 2, 2009.
http://www.digitalmediabuzz.com/2009/09/software-embraces-open-source/.

Moore, J.T.S. *Revolution OS*. http://www.revolution-os.com/about.html.

Raymond, Eric. "The Cathedral and the Bazaar." *O'Reilly*.

Riehle, Dirk. "The Economic Motivation of Open Source Software: Stakeholder Perspectives." *IEEE Computer* 40, no. 4 (April 2007).

Scacchi, Walt. "Free/open source software development: recent research results and emerging opportunities." In *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers*, 459-468. Dubrovnik, Croatia: ACM, 2007.
http://portal.acm.org/citation.cfm?id=1295019&dl=GUIDE&coll=GUIDE&CFID=60972233&CFTOKEN=54561395.

Stallman, Richard M. "Free Software, Free Society: Selected Essays of Richard M. Stallman." *GNU Press, Free Software Foundation*. www.gnupress.org.

Walker, Tom. "The Future of Open Source in Government." *gOSapps* (November 6th).